



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/902,469	07/10/2001	David S. Ebbo	MS160315.1/40062.99US01	6779

22801 7590 12/18/2003  
LEE & HAYES PLLC  
421 W RIVERSIDE AVENUE SUITE 500  
SPOKANE, WA 99201

EXAMINER
----------

STEELMAN, MARY J

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 12/18/2003

9

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/902,469

Applicant(s)

EBBO ET AL.

Examiner

Mary J. Steelman

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 07/10/01, 10/9/01, 8/29/03.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 July 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. §§ 119 and 120**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All   b) ☐ Some \*   c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
- a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 6&8.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

**DETAILED ACTION**

1. Claims 1- 6 and 8-20 are pending. There is no claim 7. The pending claims have been renumbered as 1-19 per 37 CFR 1.126.

***Information Disclosure Statement***

2. IDS received 10/09/01 and 08/29/03 has been considered.

***Drawings***

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference characters Fig. 6, "548" and "648" have both been used to designate LAN. See Specification page 27, lines 3 and 6. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference characters Fig. 6, "606" and "506" have both been used to designate system bus. See Specification page 27, lines 16 and 19. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description:

Fig. 1a: #115 is not in the Specification.

Fig. 2: #228 is not in the Specification. See page 20, line 16+.

Fig. 1b: #1612 should be #1632. See page 10, line 2.

A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

***Specification***

5. Applicant is reminded of the “Content of Specification” / (b) Cross-References to Related Applications: See 37 CFR 1.78 and MPEP § 201.11.

6. The numbering of claims is not in accordance with 37 CFR 1.126 which requires the original numbering of the claims to be preserved throughout the prosecution. When claims are canceled, the remaining claims must not be renumbered. When new claims are presented, they must be numbered consecutively beginning with the number next following the highest numbered claims previously presented (whether entered or not).

Misnumbered claims 8-20 have been renumbered 7-19.

7. Page 28, line 15, recites, “174”, should be –714--.

***Claim Objections***

8. Claim 7 (formerly Applicant’s claim 8) is objected to because of the following informalities: Claim 7 (formerly Applicant’s claim 8) recites, “A method as defined in claim 7...” Former Claim 7 does not exist. Examiner will treat this as if it reads –A method as defined in claim 6...--. Appropriate correction is required.

9. Claims 11, 12, and 13 (Formerly Applicant’s claims 12, 13, and 14) are objected to because of the following informalities: Claim 11, page 35, line 18, claim 12, page 37, line 19 and claim 13, page 38, line 19 recite “for the web page for the web page”, should be –for the web page--. Delete the repeated phrase. Appropriate correction is required.

10. Claim 17 (Formerly Applicant’s claim 18) is objected to because of the following informalities: Claim 17, page 41, line 1, recites, “comprises further comprises”, should be –

Art Unit: 2122

further comprises--. Delete the first occurrence of 'comprises'. Appropriate correction is required.

11. Claim 19 (Formerly Applicant's claim 20) is objected to because of the following informalities: Claim is dependent upon itself. Recites "as defined in claim 20", should be --as defined in claim 18--. (Claim 18 is Applicant's former claim 19.) Appropriate correction is required.

### ***Claim Rejections - 35 USC § 112***

12. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

13. Claims 14-19 (Formerly Applicant's claims 15-20) are rejected under 35 USC 112 second paragraph. Claims 14, 15, 17 and 18 recite the limitation "the resource". There is insufficient antecedent basis for this limitation in the claim.

### ***Double Patenting***

14. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686

Art Unit: 2122

F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

15. Claims 1, 3-6, and 7-19 (Formerly Applicant's claims 8- 20) are provisionally rejected under the judicially created doctrine of double patenting over claims 1-6, and 9-20 of copending Application No. 09/573768. This is a provisional double patenting rejection since the conflicting claims have not yet been patented.

The subject matter claimed in the instant application is fully disclosed in the referenced copending application and would be covered by any patent granted on that copending application since the referenced copending application and the instant application are claiming common subject matter, as follows:

In Patent Application 09/573768:

Claims 1, 10, 11, 15 pair to claims 1, 9, 10 and 14 (Formerly Applicant's claims 1, 10, 11, and 15) of Patent Application 09/902469, as the following words are substantially the same, except that '469 contains one additional limitation:

Per claims 1, 10, 11, and 15:

Art Unit: 2122

- receiving a request from the client specifying a dynamic web page content file;
- processing the dynamic web page content file to produce a source code file containing source code that represents control objects declared in the web page content file;
- compiling the source code file to produce a class from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display.

Application 09/902469 additionally includes the following limitation: “wherein the source code file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects”

Claim 2 pairs to claim 3 of Patent Application 09/902469, as the following words are substantially the same, “-the dynamic web page content file is a server-side declaration data store.”

Claims 3 and 4 pair to claims 4 and 5 of Patent Application 09/902469, as the following words are substantially the same, “the class is stored in cache memory / magnetic storage medium on the server computer system and is available to instantiate objects in response to another request specifying the dynamic web page content file.”

Claim 9 pairs to claim 8 (Formerly Applicant’s claim 9) of Patent Application 09/902469, as the following words are substantially the same, “-prior to the step of processing the dynamic web page content file, determining whether the class related to the received request has been compiled and stored in memory;

Art Unit: 2122

-if the class has been compiled and stored in memory, skipping the processing step, otherwise continue with the processing step.”

Claims 12, 13, 14 pair to claims 11, 12, and 13 (Applicant’s former claims 12, 13, and 14) of Patent Application 09/902469, as the following words are substantially the same, except that ‘469 contains one additional limitation:

Per claims 12, 13, and 14:

- receiving a request from the client computer system for the web page, wherein the request identifies a dynamic web page content file;
- creating a data model to store elements of the dynamic web page content file;
- generating a source code file related to the dynamic web page content file based on the evaluation of the data model;
- compiling the source code file to create a compiled class in memory;
- returning a class reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;
- rendering the dynamic web page content into a web page response for delivery to the client computer system;
- conducting the web page response to the requesting client computer system;
- receiving a second request for the web page wherein the request identifies a dynamic web page content file;
- determining that a compiled class for that dynamic web page content file resides in memory;



Art Unit: 2122

- returning a class reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;
- rendering the dynamic web page content into a second web page response;
- conducting the second web page response to the requesting client computer system.

Claims 5, 6, 16, 17, and 19 pair to claims 6, 7, 16, and 18 (Formerly Applicant's claims 6, 8, 17, and 19) of Patent Application 09/902469, as the following words are substantially the same, reciting a first phase, second phase and third phase (first, second, third analysis) of generating / writing source code (or intermediate data structure).

Claim 16 pairs to claim 15 (Formerly Applicant's claim 16) of Patent Application 09/902469, as the following words are substantially the same, "-parsing the resource to separate the resource into logical elements...; - creating a plurality of hierarchically related data structures ...; -storing portions of the resource in the data structures."

Claim 18 pairs to claim 17 (Formerly Applicant's claim 18) of Patent Application 09/902469, as the following words are substantially the same, "-generating an intermediate data structure, wherein the source code is generated from the intermediate data structure."

Claim 20 pairs to claim 19 (Formerly Applicant's claim 20) of Patent Application 09/902469, as the following words are substantially the same, "-the intermediate data structure is a generic description that may be translated into a plurality of source code language files, wherein at least one source code file is different from another source code language file."

Furthermore, there is no apparent reason why applicant would be prevented from presenting claims corresponding to those of the instant application in the other copending

Art Unit: 2122

application. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

***Claim Rejections - 35 USC § 102***

16. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

17. Claims 1-6 & 7-19 (Formerly Applicant's claims 8-20) are rejected under 35

U.S.C. 102(e) as being anticipated by US Patent Application 2003/0074634 by Emmelmann .

Per claim 1:

- receiving a request from the client specifying a dynamic web page content

file; (See fig. 8 (32), and page 4, [0080], "When a client browser requests a component page...")

- processing the dynamic web page content file to produce a source code

file containing source code that represent control objects declared in the web page

content file; (See fig. 8, (37) & (38), "A heitml page is seen as a program and is processed in textual order.")

-compiling the source code file to produce a class from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display;

(Page 3, [0071], "Components can be nested (hierarch) inside each other...", page 16, [0352],

Art Unit: 2122

“Inheritance is the major object oriented concept used for achieving reusability. Through inheritance it is possible to create several similar objects without specifying each one from scratch. Classes can inherit the methods of another class.” Also, [0073], “By adding components to a page, the page becomes a dynamic page or server side application.” Also, page 4, [0090], “Each component can have one, zero, or multiple component instances that are actually displayed.”)

-wherein the source code file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects. (Page 4, [0084], “Feature 2 is that components can be nested, i.e., they can contain other components. Syntactically, this is expressed by HTML/XML (source code) elements with end-tag. The content, i.e., the text between the tag and the end-tag, typically consists of browser code and other components (one or more additional dynamic web page content files). Nested components significantly improve flexibility...”)

Per claim 2:

-source code file may be referenced by a second source code file as one of the one or more additional dynamic web page content files. (Page 4, [0080], “The ISSC processor program then reads the component page and generates a browser page (source code) by using the component classes”. Also, page 4, [0084], “...components can be nested (references by a second source code file) , i.e., they can contain other components...”)

Art Unit: 2122

Per claim 3:

-the dynamic web page content file is a server-side declaration data store. (Se fig. 7, server computer with components (27), Web server software receives request, calls the appropriate page generating program (generates dynamic web page content) Also see page 6, [0122], “The ISSC processor then reads the component page (usually from a file, but it could also be stored in a database (data store)...)

Per claims 4 and 5:

-the class is stored in cache memory / on a magnetic storage medium on the server computer system and is available to instantiate objects in response to another request specifying the dynamic web page content file. (See fig. 7, (27) and page 5, [0105], “To store the component objects between page accesses, ISSC’s are based on a programming language/web server/application server that keeps the object available between different page requests.”)

Per claim 6:

-parsing the dynamic web page content file to store portions of the file into a data model, the data model comprises a plurality of data objects linked in a hierarchical manner; (See fig. 8, “parse” @ (35).)

-generating source code related to declaration information based on an analysis of the data model during a first phase;

-writing the source code related to declaration information to the source code file;

Art Unit: 2122

- generating source code related to control object information based on an analysis of the data model during a second phase;
- writing the source code related to control object information to the source code file during the second phase;
- where the one or more of the plurality of data objects correspond to compiled versions of one or more additional dynamic web page content file.

(See fig. 8, “parse” @ (35), Component editor (36) is used for component processing (37) and page generation (38). Also see page 6, [0115], “Fig. 8 is a flow chart describing what happens and in what order...The browser sends a request to the web server. The web server receives and analyzes the request, and when a component page is requested, the web server calls the component processor.” And [0129], “If a node represents browser code, it is sent to the browser unchanged. If a node represents a component, the attributes of the component are evaluated. Then, the component class of the component is identified...The display method of the component is called...Processing continues...” Also see page 17, [0371] “...insert the content of a variable into the document, and to assign some generated text to a variable...” and [0375], “The object data type (control object) is most interesting in heitml since it provides the full functionality of associative arrays. It covers the record/struct/object datatype as well as the array datatype of other languages...”)

Per claim 7 (Formerly Applicant’s claim 8):

- generating source code related to rendering information based on an analysis of the data model during a third phase ; (See figs. 8 & 9, and page 6, [0129], “Fig. 9 is a flow chart of the page

Art Unit: 2122

generation algorithm. It is a recursive algorithm taking a cb-list as parameter. The cb-list is processed node by node. If a node represents browser code, it is sent to the browser unchanged. If a node represents a component, the attributes of the component are evaluated. Then the component class of the component is identified...The display method of the component is called. Processing continues...”)

-writing the source code related to rendering information to the source code file during the third phase. (Also see fig. 11 regarding rendering.)

Per claim 8 (Formerly Applicant’s claim 9):

-prior to the step of processing the dynamic web page content file, determining whether the class related to the received request has been compiled and stored in memory; (See fig. 35, (361) regarding persistent components.)

-if the class has been compiled and stored in memory, skipping the processing step, otherwise continue with the processing step. (Fig. 35, (362) or (363).)

Per claims 9 and 10 (Formerly Applicants claims 10 and 11):

-receiving a request from the client specifying a dynamic web page content file; (Page 5, [0099], “...causes the browser to send a page request to the server...” and page 6, [0122], “The web server passes the file name...of the component page to the ISSC processor. The ISSC processor then reads the component page...and parses it. This means it finds out what tags on the page mark components...”)

Art Unit: 2122

-processing the dynamic web page content file to produce a source code file containing source code that represent control objects declared in the web page content file; (Page 5, [0099], “The ISSC technique then makes sure that the right component method is called to process (produce source code) the user interaction.”)

-compiling the source code file to produce a class from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display; (See fig. 9.)

-wherein dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional web page content files correspond to a single hierarchical object within the set of hierarchical objects. (As an example: Page 5, [0100], “...interacting with a form filed component (dynamic web page content file) does not immediately result in a server action. Instead, all information entered is collected by the browser and set to server completely as the form data set when a submit button is pressed. Therefore, the form data set can contain information for several components and trigger actions (reference to the one or more additional web page content files) of several components.”)

Per claims 11-14 (Formerly Applicant’s claims 12, 13, 14, and 15):

-receiving a request from the client computer system for the web page, wherein the request identifies a dynamic web page content file; (Page 5, [0099], “...causes the browser to send a page request to the server...” (receiving a request from the client computer) and page 6, [0122], “The web server passes the file name (identifies a dynamic web page content file)...of the

Art Unit: 2122

component page to the ISSC processor. The ISSC processor then reads the component page...and parses it. This means it finds out what tags on the page mark components...”)

-creating a hierarchical data model containing one or more control objects to store elements of the dynamic web page content file; (Page 5, [0111], “The preferred embodiment of the ISSC invention is implemented in the heitml programming language which is a page template language as well. In this embodiment, heitml is used for the component pages and heitml features are used to mark components.” And page 6, [0122-0123], “The web server passes the file name of the component page to the ISSC processor. The ISSC processor then reads the component page...and parses it...The result is an abstract syntax tree (AST). The AST contains two kinds of nodes, one for browser code and the other one for components marked on the page...The page is represented as a list of nodes called cb-list...the cb-lists connect all the children of a node or of the tree root.”)

-generating a source code file related to the dynamic web page content file based on the evaluation of the data model; (See fig. 9 and page 6, [0129] for details regarding page generation. “The cb-list is processed node by node...”)

-compiling the source code file to create a compiled class in memory; (See fig. 9.)

-returning a class reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content; (See fig. 8, (39).)

-rendering the dynamic web page content into a web page response for delivery to the client computer system; (See fig. 8, (40).)

-conducting the web page response to the requesting, client computer system; (See fig. 8, (40).)



Art Unit: 2122

-receiving a second request for the web page for the web page, wherein the request identifies a dynamic web page content file; (See fig. 8, (43), (31) and (32), "Wait for user interaction." (a second request) )

-determining that a compiled class for that dynamic web page content file resides in memory; (See fig. 11 and fig. 35. Item (71) of figure 11 can be replaced by the code of fig. 35 to handle persistent components.)

-returning a class reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content; (See fig. 8.)

-rendering the dynamic web page content into a second web page response; (See fig. 8 and fig. 35. User interaction allows for a second request to cause a second web page response.)

-conducting the second web page response to the requesting client computer system; (See fig. 8, (40).)

-wherein the dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects. (As an example: Page 5, [0100], "...interacting with a form filed component (dynamic web page content file) does not immediately result in a server action. Instead, all information entered is collected by the browser and set to server completely as the form data set when a submit button is pressed. Therefore, the form data set can contain information for several components and trigger actions (reference to the one or more additional web page content files) of several components.")

Per claim 15 (Formerly Applicant's claim 16) :

-parsing the resource to separate the resource into logical elements and identify relationships between the logical elements; (Page 2, [0025], "ISSC's (resource / components) can be reused."

And [0026], "The ISSC technique remembers information on each ISSC during dynamic page generation on the server...On the server, the event is then passed (parse until find appropriate object) to the corresponding ISSC object. The ISSC algorithm makes sure that all components work without disturbing (identifies relationships) each other, even when nested or combined."

-creating a plurality of hierarchically related data structures forming a hierarchical data model; ("The preferred embodiment of the ISSC invention is implemented in the heitml programming language which is a page template language as well. In this embodiment, heitml is used for the component pages and heitml features are used to mark components." And page 6, [0122-0123], "The web server passes the file name of the component page to the ISSC processor. The ISSC processor then reads the component page...and parses it...The result is an abstract syntax tree (AST). The AST contains two kinds of nodes, one for browser code and the other one for components marked on the page...The page is represented as a list of nodes called cb-list...the cb-lists connect all the children of a node or of the tree root." (hierarchically related data structures))

-storing portions of the resource in the data structures. (Page 6, [0123], "The AST (data structure) contains two kinds of nodes...the other one for components (resource) marked on the page."

Art Unit: 2122

Per claims 16 and 18 (Formerly Applicant's claims 17 and 19):

- performing a first analysis of the resource to generate source code / intermediate data structure elements related to variable declaration information; (An example of variable declaration information: links, buttons, fields of component [0096-0098].)
  - performing a second analysis of the resource to generate source code / intermediate data structure elements related to control object information; (An example of control object information of component: Object can program the browser [0096], object can send page request [0099].)
  - performing a third analysis of the resource to generate source code / intermediate data structure elements related to rendering information; (An example of rendering information of object: [0017], "The ISSC processor evaluates the parameters of the request and determines if the user wants to interact with any components... The result is a browser page (rendering information).)
  - storing the source code in the source code file. (Page 4, [0091], "Because of Feature 3, a component can remove its content from the page or it can display it multiple times." Also col. 8, [0164], "Clicking on the reset link, causes the same page to be redisplayed" (because it has been stored))
- (Also, see fig. 8, "parse" @ (35), Component editor (36) is used for component processing (37) and page generation (38). Also see page 6, [0115], "Fig. 8 is a flow chart describing what happens and in what order... The browser sends a request to the web server. The web server receives and analyzes the request, and when a component page is requested, the web server calls the component processor." And [0129], "If a node represents browser code, it is sent to the

Art Unit: 2122

browser unchanged. If a node represents a component, the attributes of the component are evaluated. Then, the component class of the component is identified...The display method of the component is called...Processing continues...” Also see page 17, [0371] “...insert the content of a variable into the document, and to assign some generated text to a variable...” and [0375], “The object data type (control object) is most interesting in heitml since it provides the full functionality of associative arrays. It covers the record/struct/object datatype as well as the array datatype of other languages...” Also note at page 6, [0122] – [0129], first an intermediate data structure is created (AST) then source code is generated (page generation algorithm.)

Per claim 17 (Formerly Applicant’s claim 18):

the processing step of generating source code comprises further comprises:

-generating an intermediate data structure, wherein the source code is generated from the intermediate data structure. (Page 6, [0122], “The ISSC processor then reads the component page...The result is an abstract syntax tree (AST)” (intermediate data structure). Also, [0123], “Each node representing a component with content in turn contains a cb-list that represents the content of the component. Also, [0129], “Fig. 9 is a flow chart of the page generation (source code generated) algorithm. It is a recursive algorithm taking a cb-list as parameter. The cb-list is processed node by node...”)

Per claim 19 (Formerly Applicant’s claim 20):

-the intermediate data structure is a generic description that may be translated into a plurality of source code language files, wherein at least one source code file is different from another source

Art Unit: 2122

code language file. (Page 4, [0075], "A component can be programmed in any programming language...")

*Conclusion*

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306 for regular communications and for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman



12/08/2003



**TUAN DAM**  
**SUPERVISORY PATENT EXAMINER**